

Orgs

- [Introduction](#)
- [How Does it Work?](#)
- [Creating an Organisation](#)
- [View Organisation Details](#)
- [Database Schema](#)
- [Example Database Entry](#)
- [API / Web Access](#)
 - [API Routes](#)
 - [Web Application Routes](#)

Introduction

This new access method replaces the old admin_org controller and functions.

How Does it Work?

Orgs (organisations) in Open-Audit are a key item. A user has a primary Org as well as a list of Orgs they can access. A user combines this with a list of assigned 'Roles' that define what actions they can take on items assigned to the Orgs they have access to. The combination of a users 'orgs' and 'roles' define what they can and cannot do within Open-Audit.

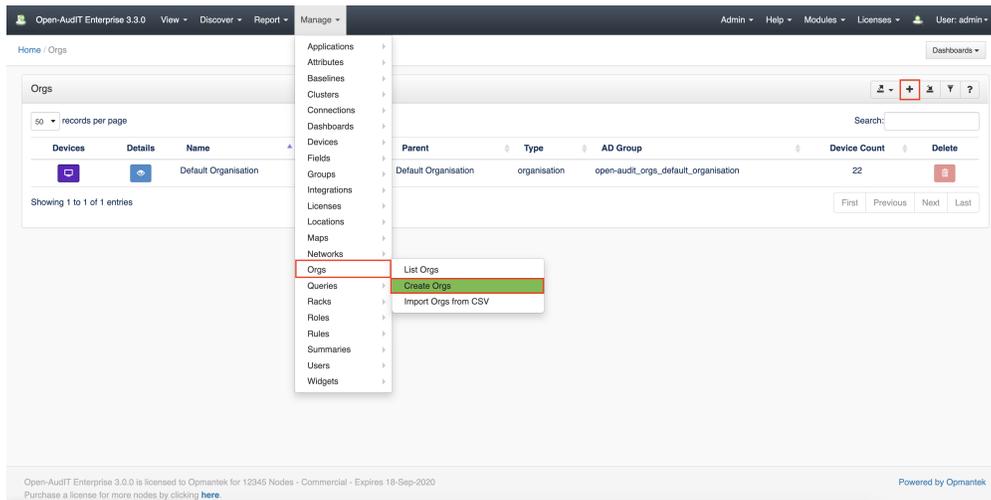
Most items in Open-Audit are assigned to an Org. Devices, Locations, Networks, etc, etc.

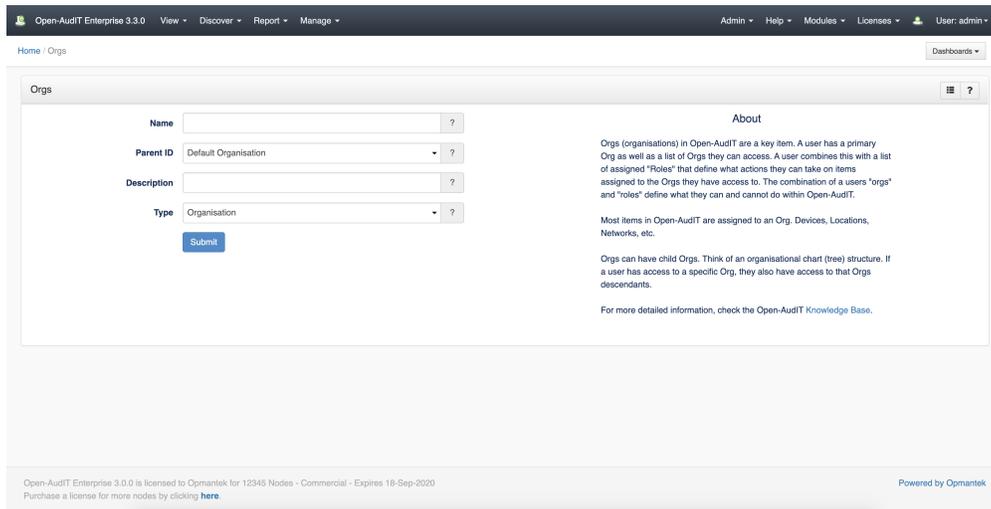
Orgs can have child Orgs. Think of an organisational chart (tree) structure. If a user has access to a specific Org, they also have access to that Orgs descendants.

Creating an Organisation

Join Paul McClendon, an Opmantek Support Engineer, as he demonstrates how to create organizations to be used in Open-Audit.

To make another organisation go to menu: Manage -> Orgs -> Create Orgs.

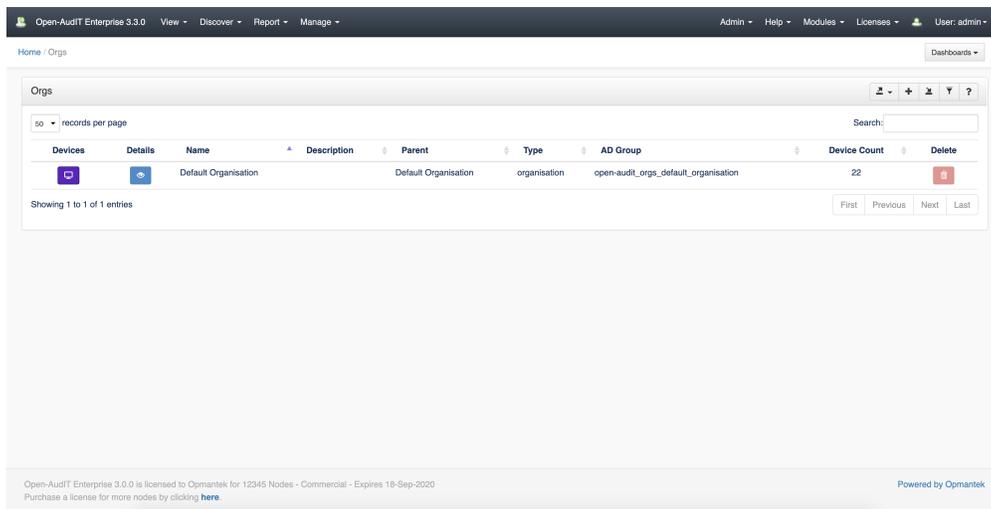




View Organisation Details

Go to menu: Manage -> Orgs -> List Orgs.

You will see a list of organisations. You can view an organisation by clicking on the blue view button. You can also edit or delete your organisations.



Database Schema

The schema for the database is below. It can also be found in the application if the user has database::read permission by going to menu: Admin -> Database -> List Tables, then clicking on the "orgs" table.

```
CREATE TABLE `orgs` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(200) NOT NULL DEFAULT '',
  `parent_id` int(10) unsigned DEFAULT '1',
  `description` text NOT NULL,
  `ad_group` varchar(100) NOT NULL DEFAULT '',
  `edited_by` varchar(200) NOT NULL DEFAULT '',
  `edited_date` datetime NOT NULL DEFAULT '2000-01-01 00:00:00',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8;
```

Example Database Entry

Orgs are stored in the database in the "orgs" table. A typical entry will look as below.

```
id: 2
name: Sea View Office
parent_id: 1
description: Awesome view
ad_group:
edited_by: Administrator
edited_date: 2017-05-17 16:31:44
```

API / Web Access

You can access the /orgs collection using the normal Open-Audit JSON based API. Just like any other collection. Please see the API documentation for further details.

API Routes

Request Method	ID	Action	Resulting Function	URL Example	Notes	Example Response
GET	n		collection	/orgs	Returns a list of organisations.	orgs_collection.json
GET	y		read	/orgs/{id}	Returns an organisation's details.	orgs_read.json
PATCH	y		update	/orgs/{id}	Update an attribute of an organisations entry.	orgs_patch.json
POST	n		create	/orgs	Insert a new organisations entry.	orgs_create.json
DELETE	y		delete	/orgs/{id}	Delete an organisations entry.	orgs_delete.json

Web Application Routes

Request Method	ID	Action	Resulting Function	URL Example	Notes
GET	n	create	create_form	/orgs/create	Displays a standard web form for submission to POST /orgs.
GET	y	update	update_form	/orgs/{id}/update	Show the networks details with the option to update attributes using PATCH to /orgs/{id}